LexGrid Vocabulary Services (LexBIG)
Use Cases

Mayo Clinic College of Medicine

Division of Biomedical Informatics

DOCUMENT REVISION HISTORY

Version Number	Date	Description
0.5	July 27, 2005	• Initial Draft

STATUS

This document is in draft form and subject to modification and additions. Specific items may still be absent or incomplete in this version.

1 Introduction

1.1 Overview

This document presents use cases for the LexGrid Terminology Services (LexBIG) project, developed by the Mayo Clinic Division of Biomedical Informatics for caBIG. The goal of the project is to build a vocabulary server accessed through a well-structured application programming interface (API) capable of accessing and distributing vocabularies as commodity resources. The server is to be built using standards-based and commodity technologies. Primary objectives for the project include:

- Provide a robust and scalable open source implementation of EVS-compliant terminology services. The API specification will be based on but not limited to fulfillment of the caBIO EVS API. The specification will be further refined to accommodate changes and requirements based on prioritized needs of the caBIG community.
- Provide a flexible implementation for terminology storage and persistence, allowing for alternative mechanisms without impacting client applications or end users. Initial development will focus on delivery of open source freely available solutions, though this does not preclude the ability to introduce commercial solutions (e.g. Oracle).
- Provide standard tooling for load and distribution of vocabulary content. This will involve support of standardized representations in UMLS Rich Release Format (RRF) and the OWL web ontology language. (Vocabulary editing, vocabulary submission, and vocabulary cross-linking are out of scope for this aim.)

1.2 Background and Scope

This document draws on use cases from the following sources:

- *Vocabulary service providers*. Describes organizations currently supporting externalized API-level interfaces to terminological content for the caBIG community. Practically speaking, this describes the NCI EVS team. Current API-level interfaces include the caCORE EVS and Metaphrase APIs. Several discussions have occurred with the EVS team to refine requirements regarding backward compatibility with existing services and data interchange formats. These requirements are formalized in the "*Consolidated server requirements*" document provided by the NCI EVS team.
- *Vocabulary integrators*. Describes organizations within the caBIG community that desire to integrate new terminological content or relations to be served to the caBIG community. This includes domain-specific stakeholders, such as the UPMC (SPIN and caTIES semantic cross-linking), University of Hawaii (food ontology), and Jackson Labs (mouse anatomy).
- *Vocabulary users*. Describes the caBIG community in general. Specific sources of information from this larger community include the V/CDE vocabulary surveys.

NOTE: Use cases described in this document are not presented as an exhaustive list for requirements definition, but rather as a tool for examining and validating main architectural pathways through the system. This is in part due to a desire to present quality over quantity but also to recognize the requirements gathering already performed by the EVS team, validated by their own first-hand experience, and delivered in the "Consolidated server requirements" document. In many ways this provides the finer grain detail, but leaves us to ensure we are still seeing the "forest through the trees".

1.3 Terminology

- For purposes of this document, 'Vocabulary' and 'Terminology' are interchangeable.

1.4 Related Documents

Description	Date / Version	Owner
LexBIG Consolidated server requirements document	July 5, 2005 / baseline	NCI
LexBIG Architecture Document	<pending>	Mayo
LexBIG Administration Guide (Install and Configuration)	<pending>	Mayo
LexBIG Programmer Guide (API and connectivity reference)	<pending>	Mayo

2 Actors

Name	Description
Application Provider	Publisher of software comprising the LexBIG vocabulary server.
caTIES Program	Retrieval application for querying pathology reports.
Client Program	An application utilizing LexBIG vocabulary services, either through direct discovery/invocation or via the caCORE EVS API.
Content Provider	Publisher of one or more specific vocabularies to be served. Publication may be in compatible native (e.g. RRF, OWL-DL) or pre-processed (e.g. LexGrid-XML) formats.
History Manager	A conceptual interface encapsulating programmatic access to history and version information maintained in the LexBIG repository.
LexBIG User / Public	Represents any user accessing the LexBIG runtime environment.
LexBIG Vocabulary Distributor	A person or institution that maintains a LexBIG server node on the grid. This server carries a registered copy of one or more vocabularies that have been made available elsewhere on the grid. Because there are no vocabularies that have LexBIG as the primary format, a LexBIG Vocabulary publisher is both a type of Vocabulary Publisher and a Vocabulary Redistributor.
License Manager	A conceptual interface encapsulating programmatic access to information describing licensing, copyright, and terms of use for vocabularies maintained in the LexBIG repository.
Research Investigator	A person active in clinical or basic science research.
Session Manager	A conceptual interface encapsulating programmatic access to LexBIG sessions, which provide a shared context across one or more bound requests to the LexBIG runtime.
System Administrator	The system administrator is responsible for installation, upgrades, and on-site monitoring of the vocabulary service software.
Vocabulary Publisher	A person or organization that is responsible for creating and/or publishing definitive revisions of vocabularies. Ideally, the entity that authors and maintains the vocabulary will also serve in the role of a primary publisher. An individual or organization that reformats. integrates or otherwise alters the format or content of a primary resource may also function as vocabulary publisher.
Vocabulary Redistributor	A person or organization that maintains an image of one or more vocabularies for use on a network or grid.
Vocabulary Server	The LexBIG runtime environment, alone or coupled with caCORE server.
Web User	User of applications externalized through internet browser-based interfaces.

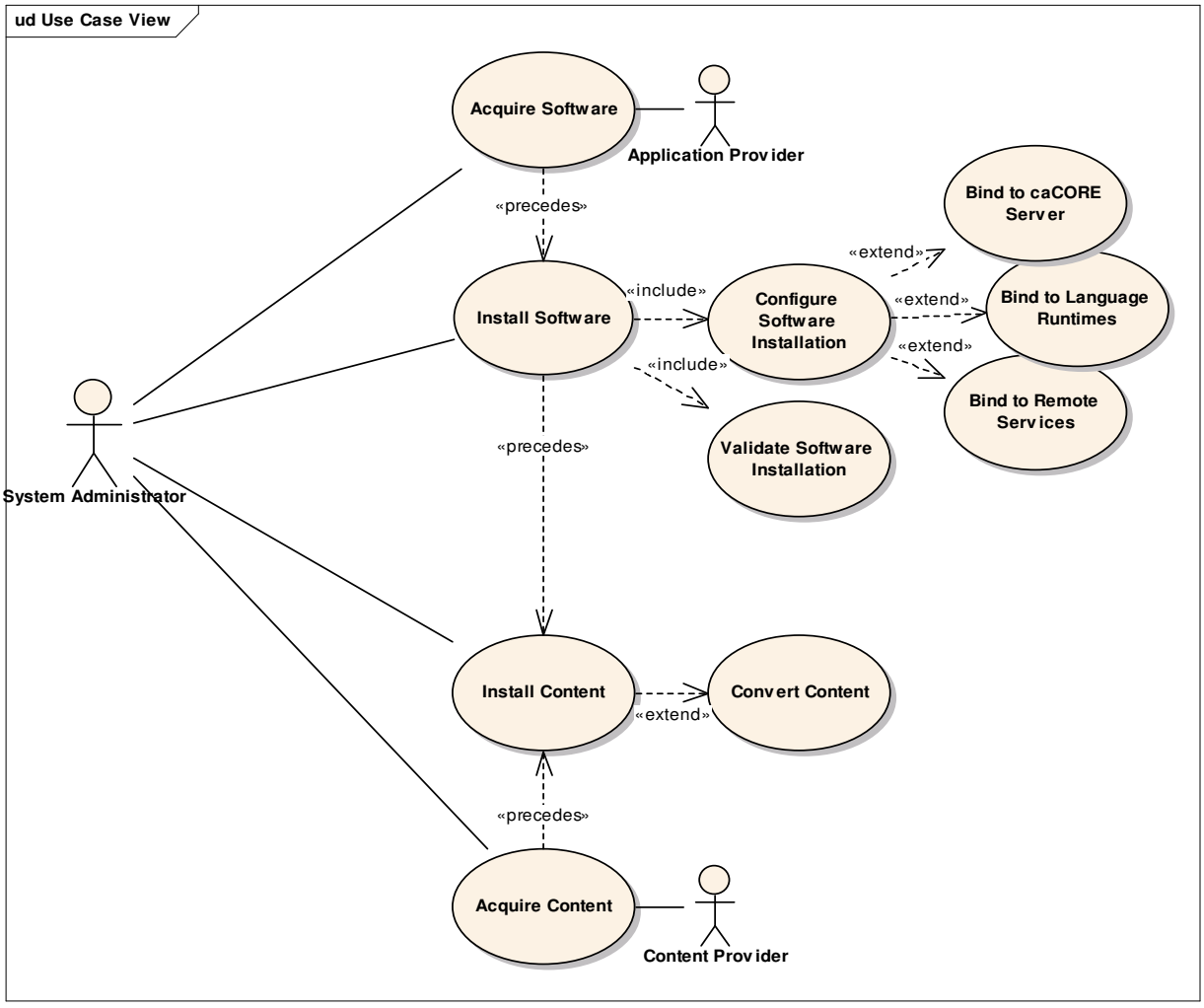
3 Use Cases

3.1 Installation

This section describes use cases to the installation of the vocabulary server and vocabulary content. These use cases support the local and federation goals of the caBIG LexGrid project. Installation of server and content provide caBIG community members the ability to deploy a local instance of the vocabulary server and content. Local installation may be desired for performance, accessibility, and control reasons. Individual caBIG community members may choose to install/host a caBIG approved vocabulary and reference different sites for other vocabularies. Upon successful, installation application developers will be able to use the caCORE SDK which bind to LexGrid API or access LexGrid API directly to support vocabulary access and usage.

3.1.1 Install Software

Install software provides the general use cases for acquiring, installing, configuring, and validating software installation. Multiple instances of a vocabulary server with the same software version or different software versions are both valid scenarios that individual caBIG members may configure.



3.1.1.1 Acquire Software

Use Case ID	INS_UC_01
Primary Actor	System Administrator
Secondary Actors	Application Provider
Brief Description	Actor receives software installation package from application provider.
Pre-conditions	Internet connectivity
Flow of Events	1) Actor selects link to download software from provider web page. 2) System presents option to register use and contact information for future updates. 3) Actor provides requested information or bypasses registration. 4) System presents license terms, requests confirmation. 5) Actor confirms the download. 6) System transmits software. 7) System logs download date/time and other statistics.
Post Conditions	Actor has local copy of software installation package.
Notes	

3.1.1.2 Install Software

Use Case ID	INS_UC_02
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Actor installs and verifies the LexBIG runtime environment.
Pre-conditions	Actor has obtained install package from application provider. Target system meets minimum hardware requirements (described in admin guide). Prerequisite software is installed and configured (described in admin guide).
Flow of Events	1) Actor opens and steps through installation documentation/instructions. 2) System provides guided installation using semi-automated programs/scripts to assist with file placement, configuration settings, etc.
Post Conditions	Runtime environment is active and prepared for installation of specific vocabularies.
Notes	Includes • Configure Software Installation (INS_UC_03) • Validate Software Installation (INS_UC_07)

3.1.1.3 Configure Software Installation

Use Case ID	INS_UC_03
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Actor customizes the LexBIG runtime installation based on preference and target environment.
Pre-conditions	See Install Software (INS_UC_02) pre-conditions.

Flow of Events	1) Actor proceeds with configuration steps defined in installation instructions. 2) System prompts actor for repository configuration (e.g. database name and connectivity settings, etc) 3) Actor provides required settings and indicates to proceed. 4) System presents actor with option to bind the LexBIG runtime to a caCORE server installation to fulfill EVS API requests. 5) Actor optionally proceeds with configuration as described by extension point Bind to caCORE Server (INS_UC_04). 6) System presents user with option to bind the LexBIG runtime to programming language runtime environments. 7) Actor optionally proceeds with configuration as described by extension point Bind to Language Runtimes (INS_UC_05). 8) System presents user with option to bind the LexBIG runtime to remote-accessible services. 9) Actor optionally proceeds with configuration as described by extension point Bind to Remote Services (INS_UC_06). 10) System centrally records all configuration settings, when handled programmatically, in anticipation of potential serviceability and uninstall needs.
Post Conditions	LexBIG runtime is installed and configured, but not verified.
Notes	System options will be presented and carried out via documentation or a semi-automated install program.

3.1.1.4 Bind to caCORE Server

Use Case ID	**INS_UC_04**
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Binds the LexBIG runtime to an existing installation of the caCORE server.
Pre-conditions	LexBIG runtime files are installed. A caCORE server is installed and properly configured on the target system.
Flow of Events	1) Actor chooses to bind the LexBIG runtime to caCORE server. 2) System requests location of caCORE server installation. 3) Actor provides location and indicates to proceed. 4) System modifies caCORE configuration as required to establish linkage (tbd) System takes any action necessary to cause changes to take effect (e.g. restart the server).
Post Conditions	EVS API requests are fulfilled through the LexBIG runtime.
Notes	Exact nature of linkage from caCORE server to LexBIG runtime services to be determined per completion of LexBIG architecture (pending). This use case is currently written assuming installation of the caCORE server and LexBIG runtime on the same machine. If required, remote access to the LexBIG runtime may also be considered and linked through remote services (see Bind to Remote Services, INS_UC_06).

3.1.1.5 Bind to Language Runtimes

Use Case ID	**INS_UC_05**
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Optimizes access to the LexBIG runtime for programs deployed on the same system.
Pre-conditions	LexBIG runtime files are installed.

Flow of Events	1) Actor chooses to bind the LexBIG runtime to language-specific environments. 2) System presents actor with supported choices (e.g. Java) 3) Actor makes selection and indicates to proceed. 4) System prompts actor for any additional required information (language-dependent). 5) Actor provides requested information, if any, and indicates to proceed. 6) System makes respective changes (e.g. adding LexBIG code to Java system classpath)
Post Conditions	LexBIG runtime API is directly accessible by the selected programming languages.
Notes	

3.1.1.6 Bind to Remote Services

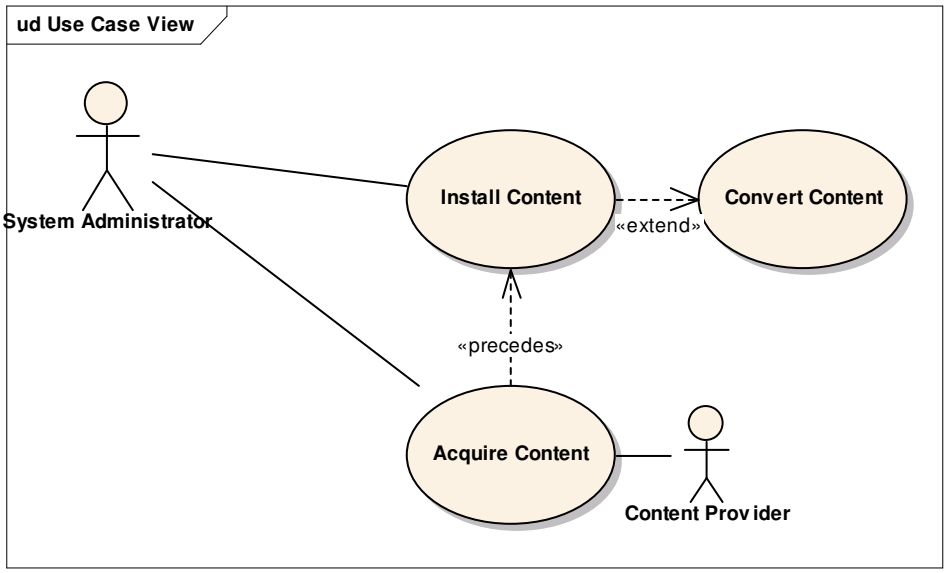
Use Case ID	**INS_UC_06**
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Registers services for subsequent discovery and invocation by client programs.
Pre-conditions	LexBIG runtime files are installed. Target service environments are active and accessible.
Flow of Events	1) Actor chooses to bind the LexBIG runtime as services for remote access. 2) System presents user with supported options (e.g. SOAP-based access through local web service container, registration against a centrally maintained vocabulary service index, registration as caGRID services, etc). 3) Actor selects service option(s) as target(s) for installation. 4) System prompts user for any required info (e.g. a web server port). 5) Actor provides requested information and confirms installation. 6) System carries out service configuration.
Post Conditions	Services can be discovered and invoked by Java/J2EE or .NET client programs.
Notes	Supported service architectures will be determined as architecture proceeds. The primary purpose of this use case is to provide for some type of direct service-oriented language-independent access to the LexBIG vocabulary services.

3.1.1.7 Validate Software Installation

Use Case ID	**INS_UC_07**
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Evaluates and reports installation status.
Pre-conditions	LexBIG runtime files are installed. Content repository is configured. Requested caCORE, language, and remote service bindings are configured.
Flow of Events	1) Actor initiates validation script or program. 2) System verifies content repository is accessible. 3) System installs test vocabulary to the repository. 4) System executes and evaluates result of build verification test suite. Tests are partitioned and run or ignored as appropriate based on availability of caCORE, direct language, and remote service bindings.
Post Conditions	If validation succeeds, server is available for general use. Otherwise, required information is logged and reported for analysis and serviceability.
Notes	

3.1.2 Install Content

Install content provides the general use case for acquiring, installing, or convert vocabulary content. Multiple instances of a vocabulary server with the same content or different versions of the content are both valid scenarios that individual caBIG members may configure.



3.1.2.1 Acquire Content

Use Case ID	INS_UC_08
Primary Actor	System Administrator
Secondary Actors	Content Provider
Brief Description	Actor receives vocabulary from content provider.
Pre-conditions	
Flow of Events	<provider-specific>
Post Conditions	Vocabulary files are available on the install system in a supported format.
Notes	Supported formats include RRF, OWL-DL, and LexGrid XML.

3.1.2.2 Install Content

Use Case ID	INS_UC_09
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Actor installs a new vocabulary to the content repository for subsequent access via the LexBIG runtime API and services.
Pre-conditions	LexBIG runtime is installed and verified (see INS_UC_07).
Flow of Events	1) Actor initiates LexBIG content install program. 2) Program requests location and format of the vocabulary to install. 3) Actor provides requested information and indicates to proceed. 4) Program initiates immediate load if vocabulary is provided in a directly consumable format; otherwise it first executes an intermediate conversion program (see Convert Content, INS_UC_10). 5) System evaluates success or failure of the install, and performs garbage collection for any temporary files no longer required.
Post Conditions	If installation succeeds, vocabulary is available for access via LexBIG runtime API and services. Otherwise, required information is logged and reported for analysis and serviceability.

Notes	

3.1.2.3 Convert Content

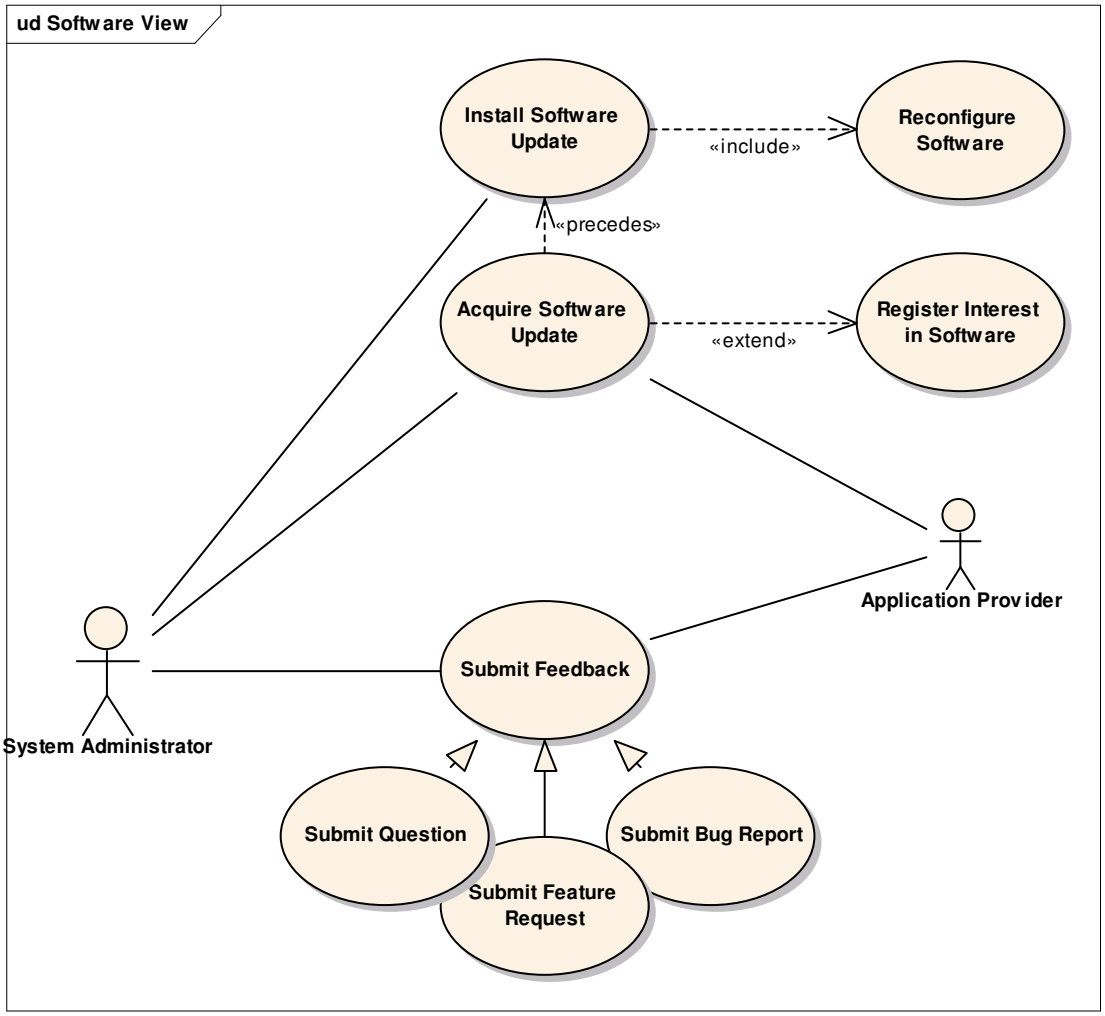
Use Case ID	INS_UC_10
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Extension point during vocabulary installation to transform content into a directly consumable format.
Pre-conditions	LexBIG runtime is installed and verified (see INS_UC_07).
Flow of Events	1) Actor initiates conversion as part of install request. 2) System utilizes LexGrid-provided conversion programs to transform content (see Content Conversion use cases)
Post Conditions	Vocabulary is available in alternate format.
Notes	

3.2 Maintenance

This section describes update and serviceability of the vocabulary server and content. Vocabulary server software and vocabulary content requires active management and maintenance. Vocabulary server software will need upgrades as functionality enhancements, bug fixes, and caCORE binding changes occur. Vocabulary content will need to be periodically updated to reflect changes in domain knowledge. Frequency of vocabulary changes will vary depending on the caBIG member or vocabulary publisher. Despite the variability of content changes the vocabulary server will provide a mechanism to support these changes.

3.2.1 Software Updates

Software update provides the general use cases for acquiring, installing, configuring, and validating new versions of the code base comprising the LexBIG runtime environment.



3.2.1.1 Acquire Software Update

Use Case ID	UPS_UC_01
Primary Actor	System Administrator
Secondary Actors	Application Provider
Brief Description	Actor receives software installation package from application provider.

Pre-conditions	Internet Connectivity
Flow of Events	1) Actor receives notification of update availability from application provider (if previously registered); otherwise actor becomes aware of update through active monitoring of provider site or other communication. 2) Actor selects link to download update from provider web page. 3) System presents option to register use and contact information for future updates. Previously registered parties can modify choice or contact information. 4) Actor provides requested information or bypasses registration. 5) System presents license terms, requests confirmation. 6) Actor confirms the download. 7) System transmits software. 8) System logs download date/time and other statistics.
Post Conditions	Actor has local copy of software update package.
Notes	

3.2.1.2 Install Software Update

Use Case ID	**UPS_UC_02**
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Actor installs the updated code and reinitializes the LexBIG runtime environment.
Pre-conditions	Actor has obtained update package from application provider.
Flow of Events	1) Actor opens and steps through update documentation/instructions. 2) System also provides guided installation using semi-automated programs/scripts to assist with file placement, configuration settings, etc. 3) System restarts any impacted servers and revalidates the installation (see Validate Software Installation, INS_UC_07). User is notified of operation success or failure. On success, system performs cleanup of temporary or unused files. On failure, applicable information is logged and reported for analysis and serviceability; actor is presented the option to revert to pre-update code base.
Post Conditions	Runtime environment is active and prepared for incoming requests.
Notes	Includes • Reconfigure Software (UPS_UC_03)

3.2.1.3 Reconfigure Software

Use Case ID	**UPS_UC_03**
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Actor customizes the LexBIG runtime installation on change to code base.
Pre-conditions	Files for updated code base are installed.
Flow of Events	1) Actor initiates update. 2) System determines whether existing configuration settings are affected. If so, system revisits applicable steps of Configure Software Installation (INS_UC_03).
Post Conditions	LexBIG runtime is installed and configured, but not verified.
Notes	

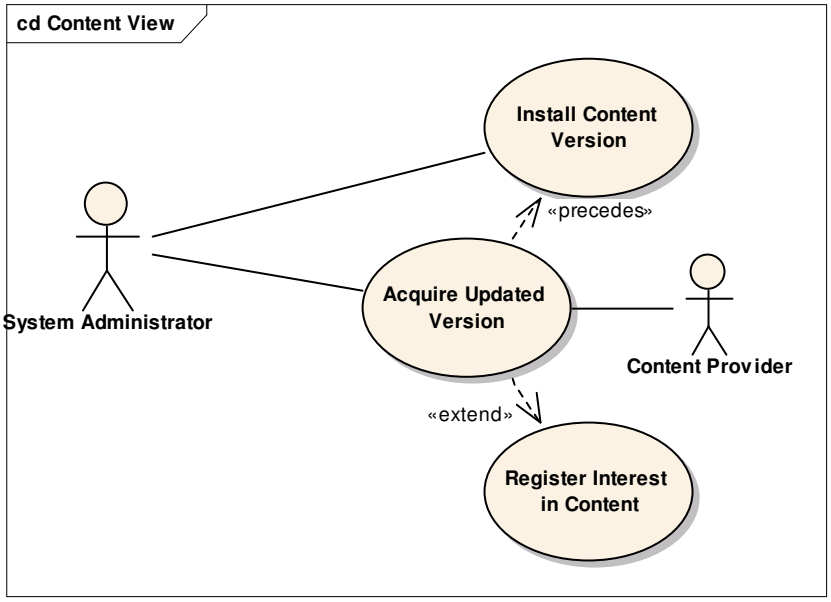
3.2.1.4 Submit Feedback

Use Case ID	**UPS_UC_04**
Primary Actor	System Administrator
Secondary Actors	Application Provider
Brief Description	Actor is provided a mechanism for ongoing dialog and issue resolution with the application provider.
Pre-conditions	Application provider has established an external presence on the internet and installed tools required to service requests from the caBIG community.

Flow of Events	1) Actor selects service tool from link on provider web site. 2) System interaction driven by web interface, specific to tools used (e.g. GForge) 3) Actor completes submission of question, bug report, or feature request. 4) System provides acknowledgement and estimated timeframe for response.
Post Conditions	Feedback delivered.
Notes	

3.2.2 Content Updates

Content update provides the general use case for acquiring and installing new versions of vocabularies already contained within the LexBIG repository.



3.2.2.1 Acquire Content Version

Use Case ID	UPC_UC_01
Primary Actor	System Administrator
Secondary Actors	Content Provider
Brief Description	Actor receives new version of vocabulary from content provider.
Pre-conditions	
Flow of Events	1) Actor receives notification of update availability from content provider (if supported by the provider and the actor has previously registered) or through other communication. 2) <Acquisition steps are provider-specific>
Post Conditions	Vocabulary files are available on the install system in a supported format.
Notes	

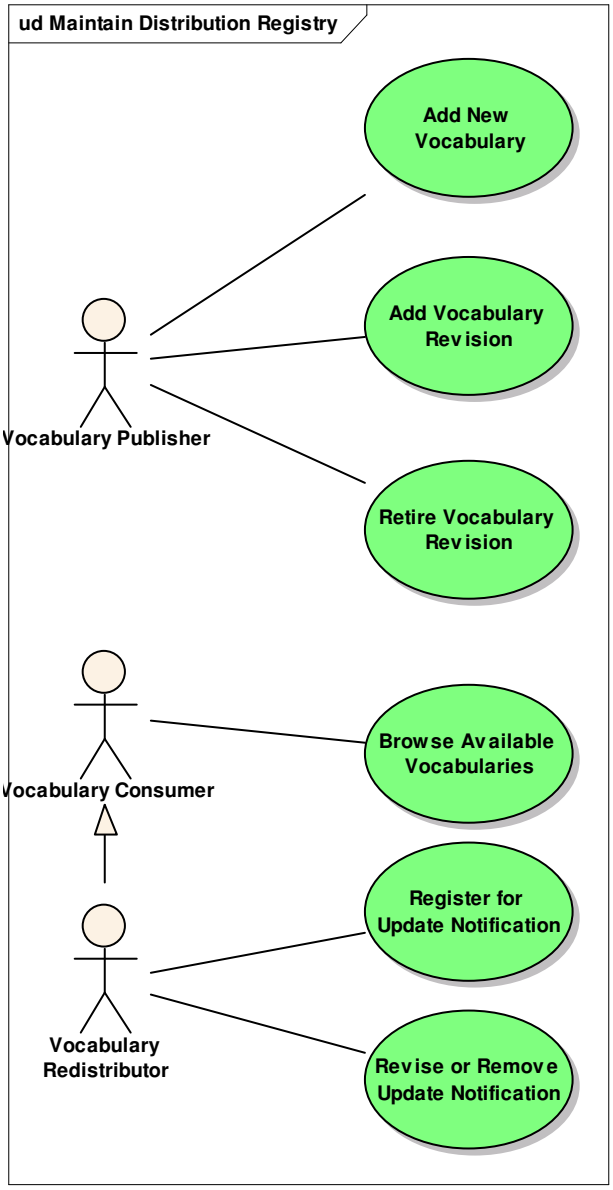
3.2.2.2 Install Content Version

Use Case ID	UPC_UC_02
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Actor installs new version of a vocabulary that pre-exists in the LexBIG repository.
Pre-conditions	LexBIG runtime is installed and verified (see INS_UC_07).
Flow of Events	1) Actor initiates LexBIG content update program. 2) System presents choice of keeping or replacing previous versions of the vocabulary. 3) Actor makes selection and indicates to proceed. 4) General steps mirroring the scratch Install Content use case, (INS_UC_09) 5) User is notified of operation success or failure. On success, system performs cleanup of temporary or unused files. On failure, applicable information is logged and reported for analysis and serviceability; actor is presented the option to revert to pre-update version of vocabulary.

Post Conditions	If installation succeeds, vocabulary version is available for access via LexBIG runtime API and services. Otherwise, required information is logged and reported for analysis and serviceability.
Notes	

3.2.3 Maintain Vocabulary Registry

Vocabulary distribution depends on a centralized resource that documents what vocabularies are available, what format they are available in and where they may be acquired. The registry also serves as a point where redistributors can register to be notified whenever the content or status of a vocabulary changes.



3.2.3.1 Add New Vocabulary

Use Case ID	REG_UC_01
Primary Actor	Vocabulary Publisher (Pub)

Secondary Actors	
Brief Description	Register the available of a new vocabulary in the registry
Pre-conditions	New vocabulary is formatted and available for use on external web site
Flow of Events	1) Publisher records description of vocabulary in registry 2) Publisher records format, download location in registry
Post Conditions	Vocabulary is registered as available for external use and upload
Notes	Registry assumes the existence of pre-established "canonical" publication formats that can be automatically parsed and unambiguously interpreted by consuming organizations. As an example, "OWL XML" is not sufficient – an exact format and semantics must be established in advance for publication in this environment.

3.2.3.2 Add Vocabulary Revision

Use Case ID	**REG_UC_02**
Primary Actor	Vocabulary Publisher (Pub)
Secondary Actors	
Brief Description	Register a new version of an existing vocabulary
Pre-conditions	A new version of a vocabulary is available for use on external web site
Flow of Events	1) Publisher records description of vocabulary revision into registry 2) Publisher records format, download location in registry 3) Publisher records format and location of formal change description in registry (where applicable) 4) Notifications are sent to all Vocabulary Redistributors who have registered an interest in this vocabulary.
Post Conditions	Vocabulary and change set are available for use and upload
Notes	Change record formats need to be in pre-established semantics and formats. These can vary from opaque blobs to (ideally) machine interpretable update records.

3.2.3.3 Retire Vocabulary Revision

Use Case ID	**REG_UC_03**
Primary Actor	Vocabulary Publisher (PUB)
Secondary Actors	
Brief Description	Record the fact that a version or versions of a vocabulary are no longer available for redistribution.
Pre-conditions	Vocabulary versions to be retired are recorded as active in the registry Vocabulary versions are available on an for use on an external web site
Flow of Events	1) Publisher records the fact that the revision is no longer available 2) Publisher records possible secondary locations where archives of the version may be located Notifications are sent to all Vocabulary Redistributors who have registered an interest in this vocabulary.
Post Conditions	Vocabulary version is marked as no longer available Vocabulary version may be removed from external web site
Notes	There also needs to be a use case for the complete removal of a vocabulary, which will include the removal of all notification requests from the registry as well.

3.2.3.4 Browse Available Vocabularies

Use Case ID	**REG_UC_04**
Primary Actor	Vocabulary Consumer
Secondary Actors	
Brief Description	Browse the vocabulary registry for documentation on published vocabularies, versions and formats.
Pre-conditions	Consumer is authorized to view registry
Flow of Events	1) Consumer uses a combination of search and browse techniques to peruse the

	vocabularies that are available for upload and use on the grid
Post Conditions	
Notes	Behavior of this use case needs to be refined at some point

3.2.3.5 Register for Update Notification

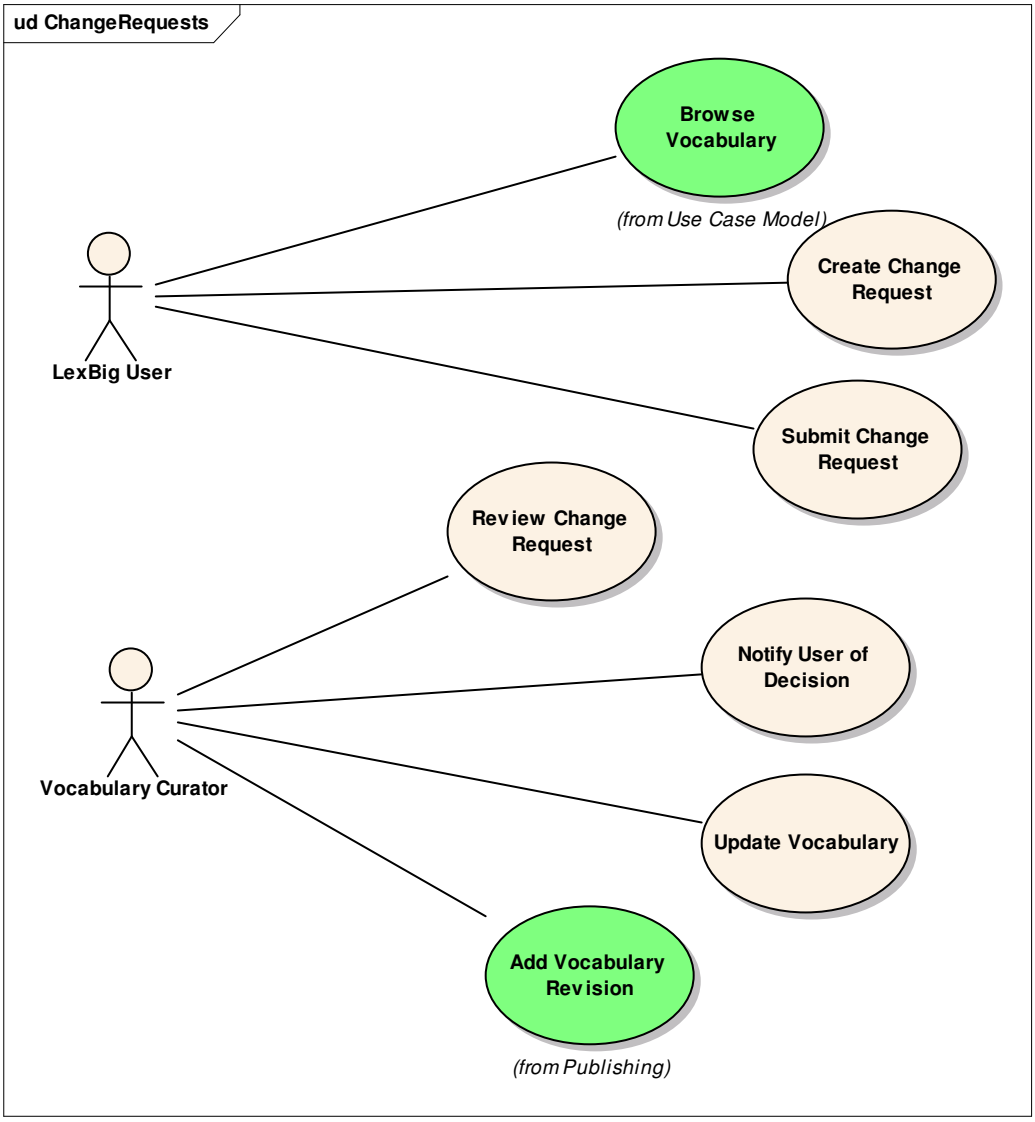
Use Case ID	**REG_UC_04**
Primary Actor	Vocabulary Redistributor
Secondary Actors	
Brief Description	Register to be notified whenever the state of a vocabulary is updated in the registry
Pre-conditions	Consumer is authorized to register for notification
Flow of Events	1) Redistributor selects a list of vocabularies for which notification is needed 2) Redistributor selects a notification mechanism (RSS, e-mail, etc.) from a list of mechanisms available on the register 3) Redistributor enters appropriate contact information for the selected mechanism 4) Redistributor enters secondary contact information (typically e-mail) for out-of-band communication. 5) An initial notification is sent to the redistributors for each of the registered vocabularies 6) Where appropriate, redistributors responds to confirm receipt of initial notification message
Post Conditions	Redistributor is registered to receive update notifications for selected vocabularies
Notes	Subsequent notifications do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.

3.2.3.6 Revise or Remove Update Notification

Use Case ID	**REG_UC_04**
Primary Actor	Vocabulary Redistributor
Secondary Actors	
Brief Description	Revise or remove a notification entry for a particular vocabulary
Pre-conditions	Redistributor or appropriate proxy (sysadmin, etc) are authorized to access registry
Flow of Events	1) Redistributor is presented list of their current notification requests 2) Redistributor selects one or more entries from the list 3) Redistributor changes appropriate notification information and/or requests to be removed from notification list 4) If primary contact information is changed, an initial notification is sent over the new primary channel and confirmation is expected where appropriate 5) If distributor is to be removed from one or more lists, a final notification event is sent via both primary and secondary channels 6) All other changes are transmitted via the secondary notification channel. Verification is expected to complete the transaction where permitted by the medium
Post Conditions	Notification records are updated appropriately
Notes	Subsequent notifications do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.

3.2.4 Content Change Requests

The LexBIG User is presented with a uniform view of all LexBIG content, so all change requests are created in terms of LexBIG structure and semantics. Using the LexBIG tooling and model, the LexBIG user makes local changes that reflect the way that the user believes that the changed vocabulary should look. The change request tooling guides the user through the process by prompting for required fields, links, etc.



3.2.4.1 Create Change Request

Use Case ID	CVL_UC_01
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	(See Create Change Request Use Case)
Pre-conditions	
Flow of Events	User creates the appropriate change request

Post Conditions	Change request is ready for submission
Notes	

3.2.4.2 Submit Change Request

Use Case ID	**CVL_UC_02**
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	User submits a change request package to the appropriate curator
Pre-conditions	User has created the necessary set of formatted change requests
Flow of Events	1) User enters description of change, urgency and secondary contact information 2) Change request package uses vocabulary registry to determine where and how to route the request 3) User is notified that request has been submitted
Post Conditions	Change request has been transmitted to the appropriate target vocabulary curator
Notes	

3.2.4.3 Review Change Request

Use Case ID	**CVL_UC_03**
Primary Actor	Vocabulary Curator
Secondary Actors	LexBIG User
Brief Description	Review a change request
Pre-conditions	Change request has been submitted by a LexBIG User
Flow of Events	1) Curator receives change request 2) Submitter is notified that request has been received 3) Curator reviews request and translates it into the semantics appropriate for the native vocabulary form 4) Curator interacts with the submitter as necessary to clarify the intent of the request
Post Conditions	Vocabulary Curator has a clear understanding of the intent and purpose of the request
Notes	

3.2.4.4 Notify User of Decision

Use Case ID	**CVL_UC_03**
Primary Actor	Vocabulary Curator
Secondary Actors	LexBIG User
Brief Description	Determine the validity and time frame for a change request and notify the requesting user
Pre-conditions	Change request is clearly understood by curator
Flow of Events	1. Curator determines whether request will be implemented, how it will be implemented and the timeframe for the release 2. Curator discusses any variations from user request with user 3. Curator sends requesting user a formal notification of the decision details, timing, etc.
Post Conditions	User and curator have agreement on what is to occur
Notes	

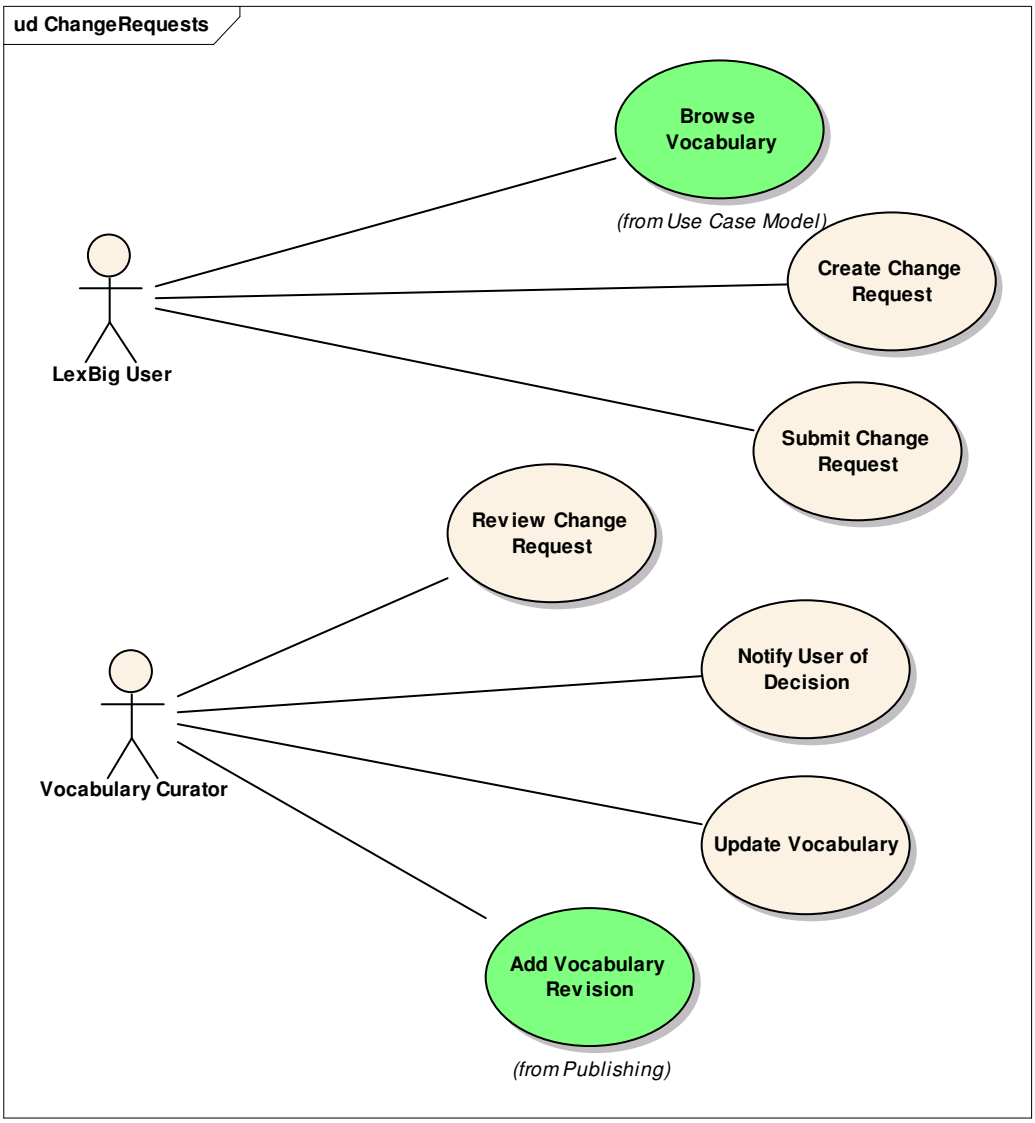
3.2.4.5 Update Vocabulary

Use Case ID	**CVL_UC_03**
Primary Actor	Vocabulary Curator
Secondary Actors	
Brief Description	Curator implements requested change in source vocabulary
Pre-conditions	User and curator have agreement on what is to occur
Flow of Events	1. Curator makes appropriate updates to target vocabulary 2. Updates are classified, validated, reviewed, etc.

	3. Revisions are converted to LexBIG form, reviewed and compared to user expectations 4. Change records are created and documentation produced
Post Conditions	Revised vocabulary is ready for publication
Notes	

3.2.5 Create Change Request

This section provides additional detail regarding the creation of change requests conforming to the structure and semantics set forth by the LexBIG model (expanded from use case CVL_UC_01 in section 3.2.4).



3.2.5.1 Modify Existing Concept

Use Case ID	CCR_UC_01
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	Make changes to attributes of an existing concept
Pre-conditions	Concept to be modified has been selected.
Flow of Events	1) User adds or removes properties, definitions, synonyms, roles, etc. on the selected properties 2) Changes are recorded as a series of "delta" records
Post Conditions	"Delta" records, when applied to the existing concept will result in a state *in the LexGrid* model that represents the desired result.

Notes	Role changes impact two or more concepts, so the target concepts will be included in the change set as well. It is quite possible that the suggested role changes will not work, but there will be no way to tell for sure until the suggested changes are converted into the native format (e.g. OWL or Ontylog) and are run through a formal classifier. As such, these are change suggestions only.

3.2.5.2 Create a New Concept

Use Case ID	**CCR_UC_02**
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	User creates a new concept as a subtype of an existing concept.
Pre-conditions	Supertype concept has been selected
Flow of Events	1) User is presented with a LexBIG template 2) User enters presentations, definitions, etc. that represent the new concept 3) User uses browser to assign any additional roles, etc. that are needed
Post Conditions	"Delta" records, when applied to the existing concept will result in a state *in the LexGrid* model that represents the desired result.
Notes	(see: CCR_UC_01)

3.2.5.3 Split a Concept

Use Case ID	**CCR_UC_03**
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	User requests that a concept be split into two new concepts
Pre-conditions	Concept to be split has been selected
Flow of Events	
Post Conditions	"Delta" records, when applied to the existing concept will result in a state *in the LexGrid* model that represents the desired result.
Notes	(see: CCR_UC_01)

3.2.5.4 Merge two Concepts

Use Case ID	**CCR_UC_04**
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	User requests that two concepts be combined into one
Pre-conditions	Concepts to be merged have been selected
Flow of Events	
Post Conditions	"Delta" records, when applied to the existing concept will result in a state *in the LexGrid* model that represents the desired result.
Notes	(see: CCR_UC_01)

3.2.5.5 Retire a Concept

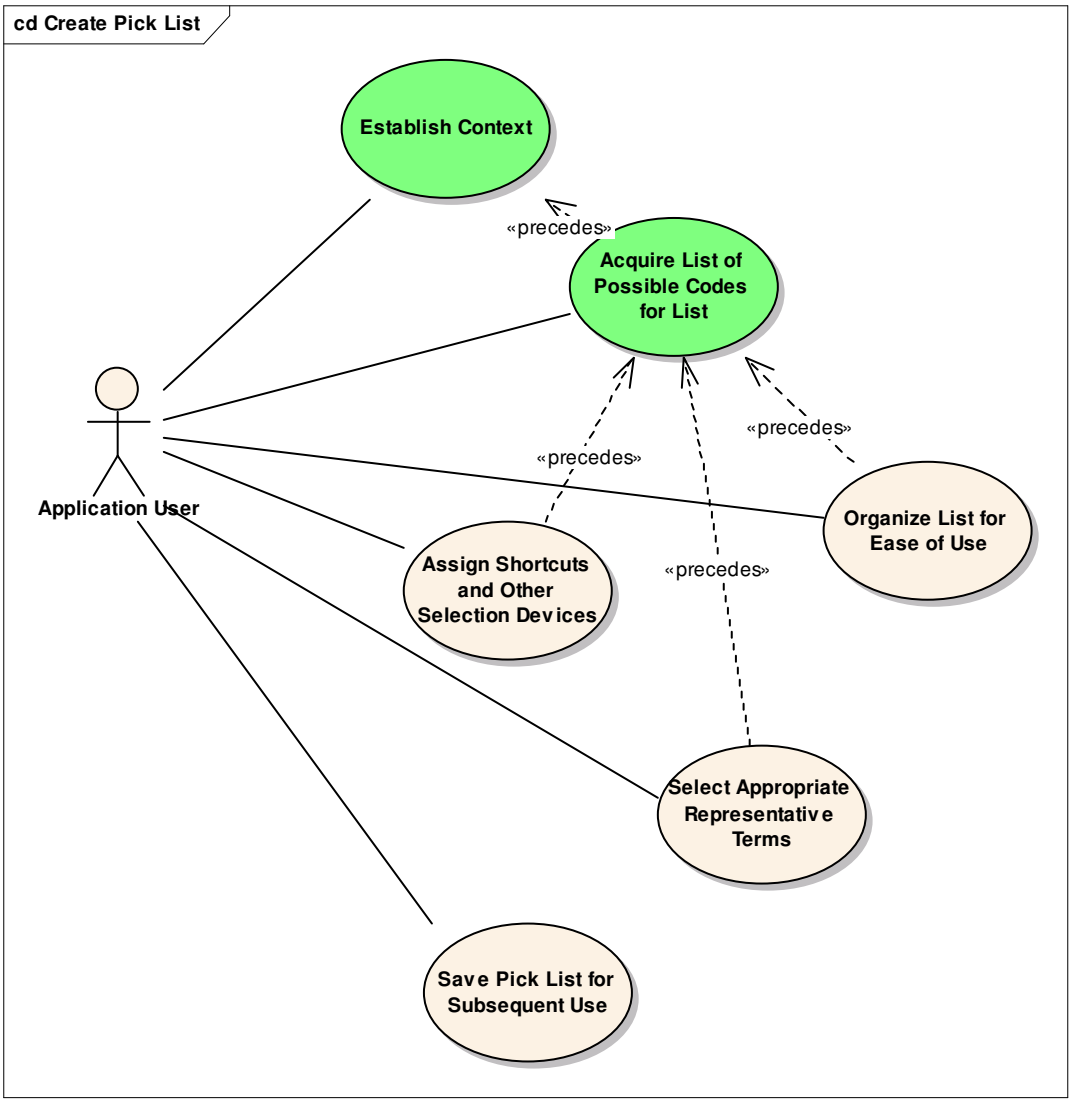
Use Case ID	**CCR_UC_05**
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	User requests that a concept code be changed from "active" to "retired"
Pre-conditions	Concepts to be retired is selected
Flow of Events	
Post Conditions	"Delta" records, when applied to the existing concept will result in a state *in the LexGrid* model that represents the desired result.
Notes	(see: CCR_UC_01)

3.2.5.6 Free text Submission

Use Case ID	CCR_UC_06
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	User submits a textual description of the set of changes that is desired
Pre-conditions	Concept to be changed has been selected
Flow of Events	User enters text that describes the change or changes desired
Post Conditions	Target concept, along with change verbiage is included as part of the change record
Notes	(see: CCR_UC_01)

3.2.6 Create Pick List

An application user organizes and labels a list of concept codes that will be use for selecting values from one or more data entry fields. While codes may not be omitted from the list, they can be re-ordered, duplicated, grouped into hierarchies and assigned the appropriate labels and, where appropriate, shortcut mechanisms for efficiency and ease of use.



3.2.6.1 Establish Context

Use Case ID	CPL_UC_01
Primary Actor	Application User
Secondary Actors	
Brief Description	The application user records the circumstances in which the pick list applies
Pre-conditions	
Flow of Events	1) A default context is acquired from the service session record 2) The user supplies • the application(s) in which this selection is applicable

	• the institutions(s), facilities, etc. where the selection applies • the individual user(s) or groups of users to which the selection applies • the desired language of the terms
Post Conditions	Context is established
Notes	

3.2.6.2 Acquire List of Possible Codes

Use Case ID	**CPL_UC_02**
Primary Actor	Application User
Secondary Actors	
Brief Description	The user selects an attribute or value domain and gets a list of enumerated values
Pre-conditions	
Flow of Events	1) User selects an attribute or enumerated value domain 2) Vocabulary server returns a list of possible codes and associated designations for the domain
Post Conditions	
Notes	

3.2.6.3 Organize list for ease of use

Use Case ID	**CPL_UC_03**
Primary Actor	Application User
Secondary Actors	
Brief Description	Restructure and reorder a list for maximum flexibility
Pre-conditions	Complete of possible values has been selected
Flow of Events	User reorders nodes, duplicates nodes, flattens hierarchies, establishes new non-selectable intermediate nodes, etc. to create a list that is optimal for quick selection
Post Conditions	List has a new hierarchy and/or order.
Notes	Vocabulary suppliers may object to the ability to restructure a list hierarchy because it adds unintended or invalid semantics to the resulting code set. If this is the case, restructuring should be discouraged or disallowed.

3.2.6.4 Select Appropriate Representative Terms

Use Case ID	**CPL_UC_04**
Primary Actor	Application User
Secondary Actors	
Brief Description	Assign “meaningful” representative terms to list nodes as needed
Pre-conditions	
Flow of Events	1) User selects an appropriate synonym from a list of possible designations of a concept code. 2) User creates a new representative term where appropriate
Post Conditions	
Notes	Vocabulary suppliers and regulatory agencies may object to the ability to assign new representative terms that aren’t a part of the actual vocabulary. When this is the case, the ability to create new representative terms should be restricted or disabled.

3.2.6.5 Assign Shortcuts and other Selection Devices

Use Case ID	**CPL_UC_5**
Primary Actor	Application User
Secondary Actors	
Brief Description	User assigns shortcuts, mnemonics and other access codes to list entries
Pre-conditions	
Flow of Events	

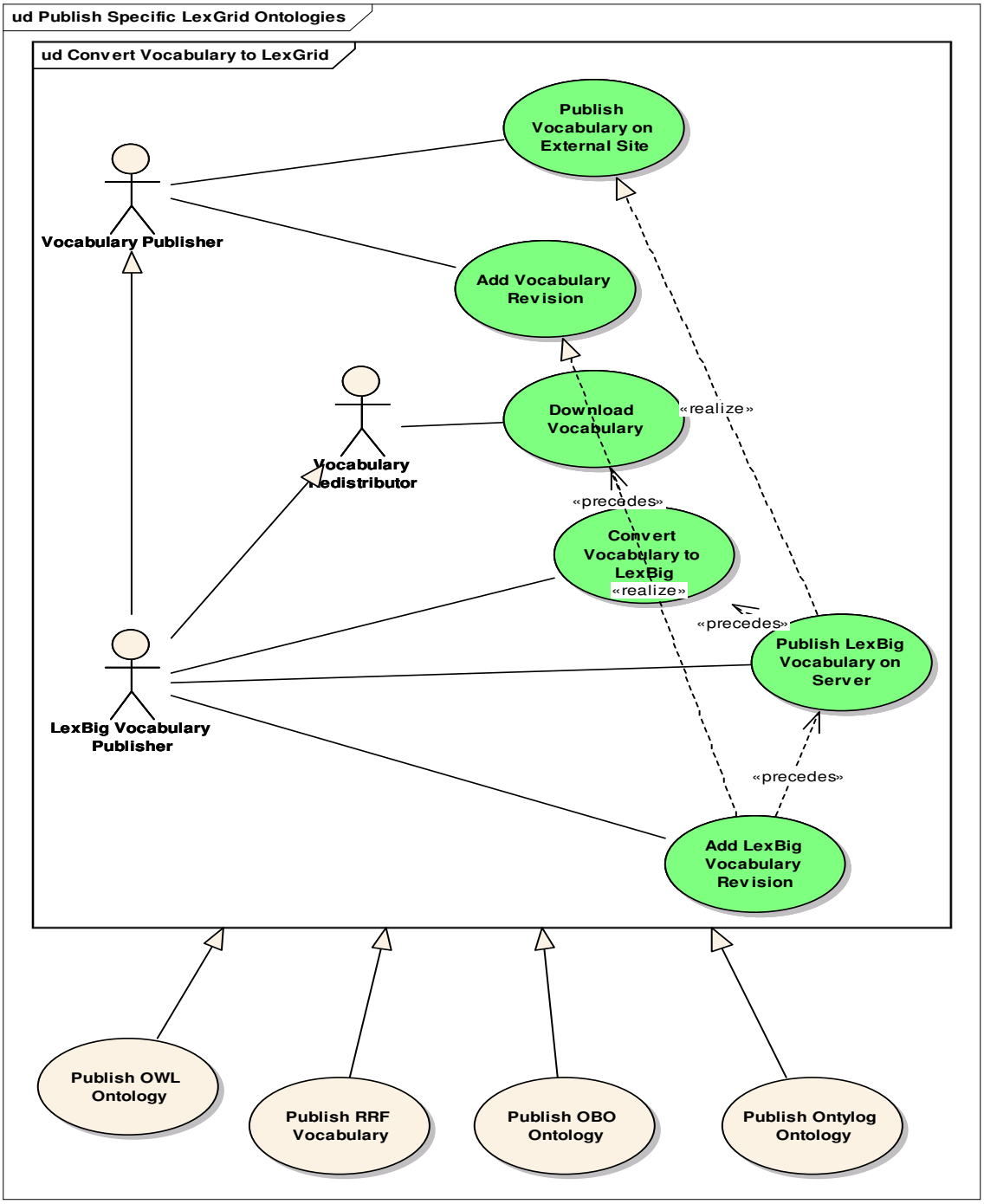
Post Conditions	
Notes	While this is an important step in application building, it tends to be platform and application specific and, as a consequence, may be out of scope.

3.2.6.6 Save Picklist for Subsequent Use

Use Case ID	**CPL_UC_5**
Primary Actor	Application User
Secondary Actors	
Brief Description	User saves resulting list in repository for subsequent use
Pre-conditions	
Flow of Events	
Post Conditions	
Notes	Distribution, naming, scope, specialization, coordination w/ vocabulary changes, etc. are all unresolved issues at this point.

3.3 Content Conversion

To be used within the LexBIG Grid, a vocabulary must be converted to a format that is accessible through the standardized service software. Any updates or revisions must be recognized, converted and synchronized as well. New LexBIG resources on the grid need to go through the same registration/notification process as any other publication format.



3.3.1 Convert Vocabulary to LexBIG

Use Case ID	CVL_UC_01
Primary Actor	LexBIG Vocabulary Publisher
Secondary Actors	
Brief Description	Convert a downloaded vocabulary into the LexBIG model
Pre-conditions	1) A new or revised vocabulary has been downloaded from a distribution center 2) The downloaded vocabulary is represented in a "canonical format" that is recognized by the conversion tools 3) The LexBIG Vocabulary Publisher has downloaded and installed the LexBIG conversion suite 4) The LexBIG Vocabulary Publisher has downloaded and installed the LexBIG server software
Flow of Events	1) Convert vocabulary into LexBIG format 2) Convert change records into LexBIG format
Post Conditions	LexBIG format of vocabulary revision is ready for upload
Notes	In many cases, this use case may merge with CVL_UC_02 because there won't be an intermediate form beyond what is installed on the server

3.3.2 Publish LexBIG Vocabulary on Server

Use Case ID	CVL_UC_02
Primary Actor	Vocabulary Publisher (Pub)
Secondary Actors	
Brief Description	Install a converted LexBIG format vocabulary onto a LexBIG grid server
Pre-conditions	LexBIG formatted vocabulary revision is ready for install
Flow of Events	1) Install converted vocabulary and changes onto local server 2) Update lexical indices to reflect changes 3) Run validation/verification tests on conversion
Post Conditions	Vocabulary is validated and ready to be officially registered on the grid
Notes	

3.3.3 Add LexBIG Vocabulary Revision

Use Case ID	CVL_UC_03
Primary Actor	Vocabulary Publisher (PUB)
Secondary Actors	
Brief Description	Register a newly available vocabulary on the grid and in the central vocabulary registry.
Pre-conditions	Vocabulary is published on a local LexBIG server
Flow of Events	1) Register converted vocabulary on lexical grid 2) Register converted vocabulary in central vocabulary registry
Post Conditions	Revised vocabulary is live on the grid Revisions have been registered and interested parties have been notified
Notes	

3.3.4 Load RRF Content into LexBIG.

Use Case Id	CVL_UC_04
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Actor loads RRF file into LexBIG server.
Pre-conditions	LexBIG servers have been installed and configured. RRF Content has been downloaded.

	LexBIG conversion tool is available.
Flow of Events	1) Actor launches the LexBIG conversion tool. 2) Actor chooses the appropriate conversion. 3) Tool requests necessary parameters for the conversion. 4) Actor enters the requested parameters, including the location of the RRF file, and the address and authentication information for their LexBIG server. 5) Actor launches the transform process. 6) Tool validates the entered parameters. 7) Tool loads the data.
Post Conditions	RRF File has been loaded into a local LexBIG server.
Notes	

3.3.5 Create LexBIG indexes for RRF Data.

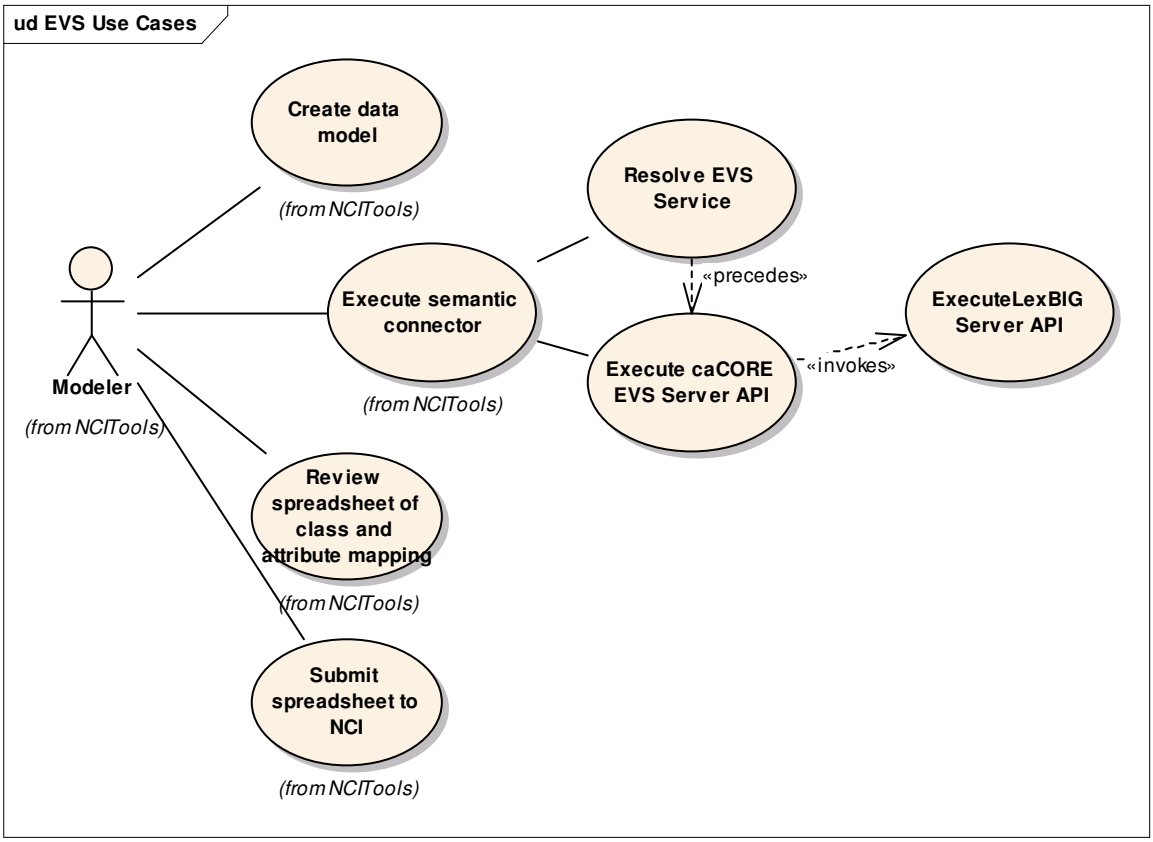
Use Case Id	**CVL_UC_05**
Primary Actor	System Administrator
Secondary Actors	
Brief Description	Actor creates LexBIG indexes for the newly loaded RRF data.
Pre-conditions	Actor has completed Conversion_01 LexBIG indexer tool is available. Optional – LVG installation is available.
Flow of Events	1) Actor launches the LexBIG indexer tool. 2) Tool requests necessary parameters for indexing. 3) Actor enters the requested parameters, including the address and authentication information for the LexBIG server, and the index output location. 4) Optional – Actor enters LVG configuration information if they wish to build a normalized index. 5) Actor launches the index process. 6) Tool validates the parameters and indexes the data. 7) Actor modifies the configuration for the LexBIG server so that it knows about the new index.
Post Conditions	Actor has constructed and registered an index on the LexBIG data with the LexBIG server.
Notes	

3.4 Runtime Access

This section describes typical runtime use cases for the vocabulary server. Runtime access includes access to server through existing NCI tools such as NCI Browser or caCORE semantic connector. Runtime access may connect through caCORE or LexGrid API.

3.4.1 EVS Access: Annotate a Data Model using Semantic Connector

A modeler (user) creates a data model representing the data elements for their application. The data model is exported using XML Metadata Interchange (XMI). The modeler executes the semantic connector software. The semantic connector uses the XMI file and attempts to annotate the UML class and attribute names to the NCI thesaurus vocabulary. A spreadsheet is created enumerating the UML class, UML entity, concept code, concept name, and concept definition. If a match is not found, the concept information is blank. The modeler can refine the data model based on annotations and rerun the semantic connector. The final spreadsheet is submitted to NCI for review.



3.4.1.1 Create data model

Use Case ID	NCI_UC_01
Primary Actor	Modeler
Secondary Actors	Modeling Tool – e.g. Enterprise Architect
Brief Description	Actor creates a data model based on the data requirements of the application or system.
Pre-conditions	Modeling tool that can generate XMI representation
Flow of Events	1) Create class (entities) e.g. Gene 2) Create attributes (columns) e.g. Identifier

	3) Create necessary relationships 4) Review model for proper normalization (3NF) 5) Save model in XML Metadata Interchange format
Post Conditions	A model saved as XMI
Notes	

3.4.1.2 Execute semantic connector

Use Case ID	**NCI_UC_02**
Primary Actor	Modeler
Secondary Actors	
Brief Description	Actor runs the semantic connector using the XMI file as input
Pre-conditions	XMI file is created. Installation caCORE SDK. Internet access
Flow of Events	1) Copy XMI file into appropriate directory 2) Run semantic connector 3) Check acknowledgement of run
Post Conditions	A spreadsheet of UML class and attributes with concept mapping
Notes	

3.4.1.3 Resolve EVS Service

Use Case Id	**EVS_UC_01**
Primary Actor	caCORE EVS User
Secondary Actors	caCORE Client
Brief Description	Actor finds and connects to a caCORE server.
Pre-conditions	caCORE has been installed and configured with a LexBIG EVS backend. caCORE connection address has been published. Actor has downloaded and installed the appropriate caCORE client application. Actor has documentation for the caCORE client application.
Flow of Events	1) Actor finds the caCORE connection address. 2) Actor enters the connection address into the caCORE client as documented by the caCORE documentation. 3) Actor executes method as specified by the caCORE documentation to connect to the caCORE server. 4) caCORE client attempts to make connection to the caCORE server. a. Success – caCORE client is ready for EVS calls. b. Failure – caCORE client returns an error message to the Actor.
Post Conditions	Actor has a connection to a caCORE server.
Notes	

3.4.1.4 Invoke caCORE EVS API

Use Case Id	**EVS_UC_02**
Primary Actor	caCORE EVS User
Secondary Actors	caCORE Client caCORE Server
Brief Description	Actor calls an EVS method from the caCORE API.
Pre-conditions	Actor has completed EVS_UC_01
Flow of Events	1) Actor chooses a method to call from the EVS caCORE API. 2) Actor supplies requested parameters and executes the chosen method. 3) caCORE Client forwards parameters and method call to caCORE server. 4) caCORE Server forwards the parameters and method call to an appropriate LexBIG method call. 5) LexBIG method call returns the result(s) or error(s) to the caCORE server.

	6) caCORE server returns result(s) or error(s) to the caCORE client. 7) The caCORE client returns the result(s) or error(s) to the Actor.
Post Conditions	Actor has constructed and registered an index on the LexBIG data with the LexBIG server.
Notes	It is assumed that the LexBIG EVS caCORE server implements all of the methods as specified by the caCORE API that are implemented in the current caCORE server. It is assumed that the input parameters and results from the LexBIG EVS caCORE server will be the same as they are in the current caCORE implementation.

3.4.1.5 Review spreadsheet of class and attribute mapping

Use Case ID	**caTIES_UC_03**
Primary Actor	Modeler
Secondary Actors	NA
Brief Description	Actor opens spreadsheet file and reviews mapping of class names and attributes. Multiple mappings require the modeler to choose one of the choices. Blank mapping requires the modeler to review class/attribute name for spelling and camel case.
Pre-conditions	Spreadsheet software capable of reading comma separated file format
Flow of Events	1) Open annotated file 2) Review attributes for multiple mappings or missing values 3) Review class/attribute naming for missing values 4) Refine model and repeat semantic connector run 5) For missing or incorrect mappings enter appropriate concept definition 6) Submit to NCI
Post Conditions	A reviewed spreadsheet file with appropriate comments
Notes	

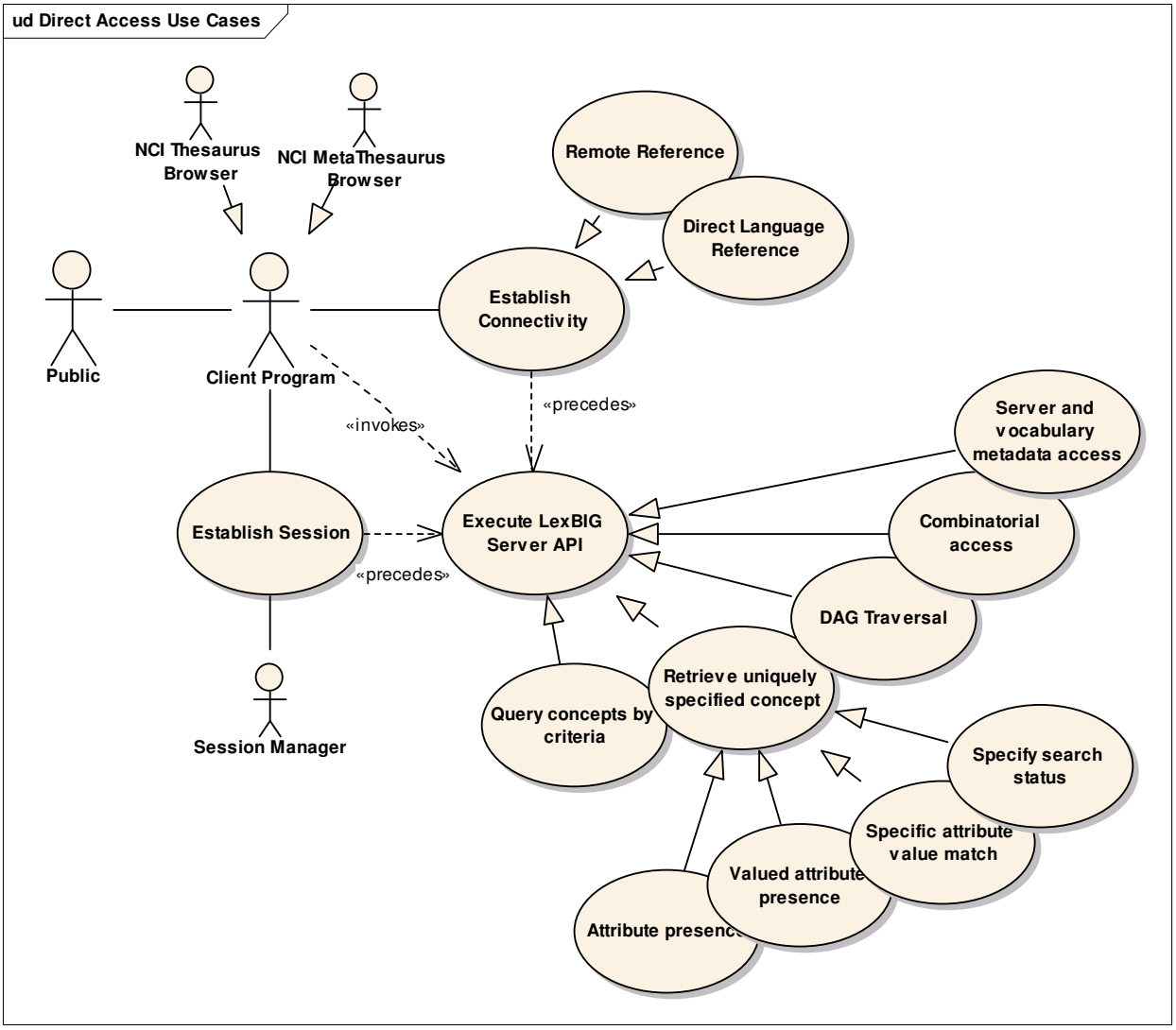
3.4.1.6 Submit spreadsheet to NCI

Use Case ID	**caTIES_UC_04**
Primary Actor	Modeler
Secondary Actors	NA
Brief Description	Actor submits model changes and mappings to NCI fore review
Pre-conditions	All class and attributes have been reviewed and checked for accuracy
Flow of Events	1) Actor submits to NCI for review 2) Actor interacts with NCI modeler for appropriate definition 3) NCI suggests mapping for missing or incorrect assignments 4) Additional vocabulary may be created based on new definitions
Post Conditions	Final spreadsheet file of class/attribute names with concept mapping
Notes	

3.4.2 Programmatic Access to LexBIG API

Programmatic access represents the primary pathway for accessing vocabulary content. This pathway can be used by a variety of external applications requiring vocabulary resources and functionality.

In addition to programmatic access through the caCORE EVS API, the LexBIG runtime environment supports a native API that may provide additional functionality not present in caCORE EVS. Invocation of the LexBIG API can be made using direct language binding or a service based method.



3.4.2.1 Establish Connectivity

Use Case Id	PGM_UC_01
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	Actor establishes a connection to a LexBIG server (remote or direct language reference)
Pre-conditions	All necessary LexBIG server components are installed, loaded, and configured. Actor has necessary LexBIG client code (client stubs for remote access, LexBIG Implementation for direct language reference).

Flow of Events	1) Actor calls method to instantiate a connection, as documented for the LexBIG client code that they are using. 2) Appropriate use case is executed for client code (either Remote Reference or Direct Language Reference).
Post Conditions	Actor has a connection to a caCORE server.
Notes	

3.4.2.2 Remote Reference

Use Case Id	**PGM_UC_02**
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	Connection to LexBIG is established to a remote server.
Pre-conditions	PGM_UC_01 Actor has requested a connection to a remote server.
Flow of Events	1) Client library attempts to connect to the remote server. 2) Client library reports successful connection, or returns error to the user.
Post Conditions	Actor has a connection to a caCORE server.
Notes	

3.4.2.3 Direct Language Reference

Use Case Id	**PGM_UC_03**
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	Direct language reference to LexBIG implementation is established. Actor has requested a connection to a remote server.
Pre-conditions	PGM_UC_01
Flow of Events	1) LexBIG implementation is initialized. 2) Implementation reports initialization success or failure to User.
Post Conditions	Actor has a connection to a caCORE server.
Notes	

3.4.2.4 Establish Session

Use Case Id	**PGM_UC_04**
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	Actor requests a session from the LexBIG server, and sets session parameter values.
Pre-conditions	Actor has established connectivity (PGM_UC_01) Session Manager is available.
Flow of Events	1) Actor calls client API method to request a session. 2) Session Manager creates a session, and returns session information to client. 3) Client stores session information. 4) Actor calls method to change a session default value. 5) Client sends stored session information and new value into the session manager. a. Session Manager validates new value. b. Session Manager stores new value if valid, or returns an error if invalid.
Post Conditions	Actor has established a session.
Notes	If the Actor is using a direct language reference, the Client is the same thing as the LexBIG API implementation. If the Actor is using a remote reference, all actions are passed from the client to the LexBIG API implementation.

3.4.2.5 Method Invocation

Use Case Id	PGM_UC_05
Primary Actor	LexBIG User
Secondary Actors	
Brief Description	Actor calls a method from the LexBIG API.
Pre-conditions	Actor has established connectivity (PGM_UC_01) Optional – Actor has established session (PGM_UC_04)
Flow of Events	1) Actor provides parameters for and calls a LexBIG API client method. 2) The client passes the parameters and method call in to the LexBIG API Implementation. 3) Optional – if session information is present, the session information is also passed in. 4) The implementation sends the parameters on to the proper method (see use cases PGM_UC_06 through PGM_UC_10 5) The method returns the result to the Actor.
Post Conditions	Actor get results from calling a method.
Notes	If the Actor is using a direct language reference, the Client is the same thing as the LexBIG API implementation. If the Actor is using a remote reference, all actions are passed from the client to the LexBIG API implementation.

3.4.2.6 Retrieve uniquely specified concept

Use Case Id	PGM_UC_06
Primary Actor	LexBIG User
Secondary Actors	Method
Brief Description	Actor calls a method from the LexBIG API to retrieve a uniquely specified concept.
Pre-conditions	Actor has invoked a method (PGM_UC_05)
Flow of Events	1) Method receives necessary parameters to get a uniquely specified concept. 2) Method validates parameters according to the LexBIG API specification (tbd) a. If parameters are invalid, an error is returned. 3) LexBIG implementation is executed to get the desired result. 4) Result is returned to the Actor.
Post Conditions	Actor gets details (as specified by the LexBIG API) of the uniquely specified concept.
Notes	

3.4.2.7 Query concepts by criteria

Use Case Id	PGM_UC_07
Primary Actor	LexBIG User
Secondary Actors	Method
Brief Description	Actor calls a method from the LexBIG API to query concepts by specified criteria.
Pre-conditions	Actor has invoked a method (PGM_UC_05)
Flow of Events	1) Method receives necessary parameters to query concepts by criteria.. 2) Method validates parameters according to the LexBIG API specification (tbd) a. If parameters are invalid, an error is returned. 3) LexBIG implementation is executed to get the desired result. 4) Result is returned to the Actor.
Post Conditions	Actor gets details (as specified by the LexBIG API) of the query.
Notes	

3.4.2.8 DAG Traversal

Use Case Id	PGM_UC_08

Primary Actor	LexBIG User
Secondary Actors	Method
Brief Description	Actor calls a method from the LexBIG API to traverse the Directed Acyclic Graph.
Pre-conditions	Actor has invoked a method (PGM_UC_05)
Flow of Events	1) Method receives necessary parameters to traverse the DAG in the requested direction, to the requested depth. 2) Method validates parameters according to the LexBIG API specification (tbd) a. If parameters are invalid, an error is returned. 3) LexBIG implementation is executed to get the desired result. 4) Result is returned to the Actor.
Post Conditions	Actor gets the requested nodes (as specified by the LexBIG API) from the DAG.
Notes	

3.4.2.9 Combinatorial Access

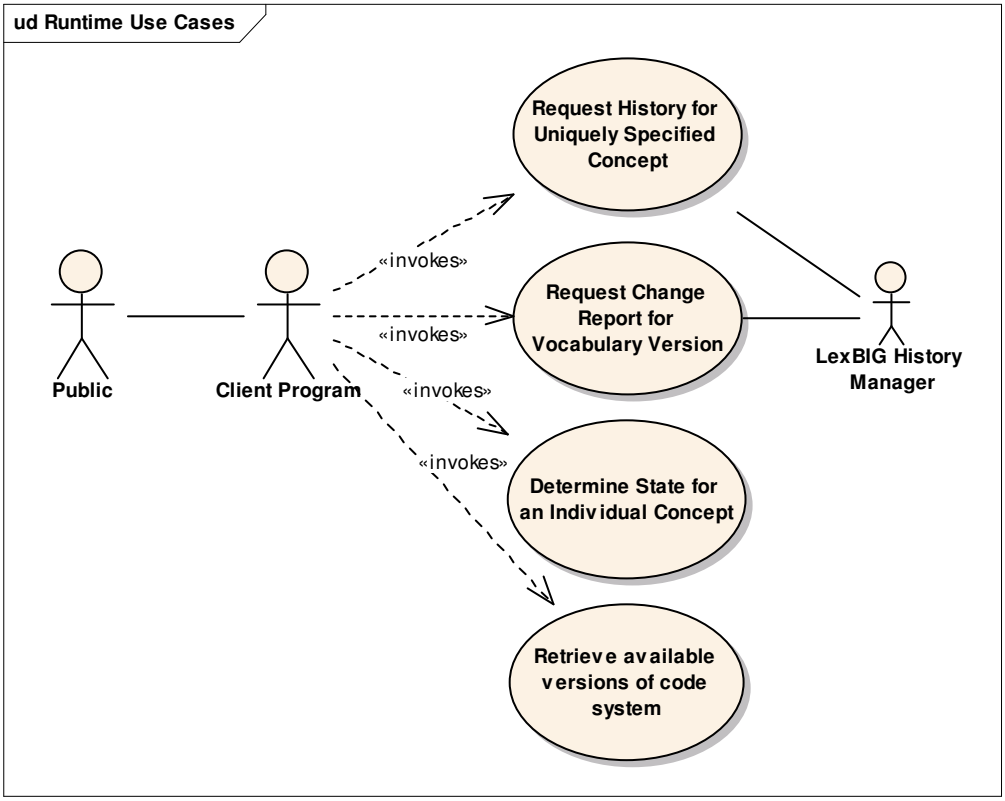
Use Case Id	**PGM_UC_09**
Primary Actor	LexBIG User
Secondary Actors	Method
Brief Description	Actor calls a method from the LexBIG API to execute a combination of other methods.
Pre-conditions	Actor has invoked a method (PGM_UC_05)
Flow of Events	1) Method receives necessary parameters to execute a combination of other methods. 2) Method validates parameters according to the LexBIG API specification (tbd) a. If parameters are invalid, an error is returned. 3) Multiple LexBIG methods are executed, using the results from one execution as input to the next method to get the desired result. 4) Result is returned to the Actor.
Post Conditions	Actor gets results (as specified by the LexBIG API) from the combination query.
Notes	

3.4.2.10 Server and Vocabulary Metadata Access

Use Case Id	**PGM_UC_10**
Primary Actor	LexBIG User
Secondary Actors	Method
Brief Description	Actor calls a method from the LexBIG API to get metadata.
Pre-conditions	Actor has invoked a method (PGM_UC_05)
Flow of Events	1) Method receives necessary parameters to get requested metadata. 2) Method validates parameters according to the LexBIG API specification (tbd) a. If parameters are invalid, an error is returned. 3) LexBIG implementation is executed to get the desired metadata. 4) Result is returned to the Actor.
Post Conditions	Actor gets results (as specified by the LexBIG API) from their metadata query.
Notes	

3.4.3 Access to History and Version Information

A number of scenarios justify the ability of a vocabulary server to provide access to vocabulary history and vocabulary versions. This functionality can be addressed with varying degrees of sophistication. Vocabulary history needs to provided and maintained by the vocabulary providers. There is no computable mechanism to determine a concept history if not provided by vocabulary provider. Vocabulary versions have many uses for information retrieval supporting epidemiologic use cases. Versioning provides the ability to determine how a concept was used in a given time period. Ultimately, versioning would be best maintained by provider, but with separate namespace individual versions of a vocabulary can be published and accessed.



3.4.3.1 Request History for Uniquely Specified Concept

Use Case Id	HST_UC_01
Primary Actor	LexBIG User
Secondary Actors	Client Program, History Manager
Brief Description	Queries historical entries for a given concept.
Pre-conditions	Historical information has been imported from native format to the LexBIG repository.
Flow of Events	1) Actor invokes LexBIG API through client program specifying concept ID 2) System queries available history based the given parameters
Post Conditions	Actor gets results (as specified by the LexBIG API) from the query.
Notes	

3.4.3.2 Request Change Report for Vocabulary Version

Use Case Id	HST_UC_02
Primary Actor	LexBIG User
Secondary Actors	Client Program, History Manager

Brief Description	Queries historical entries recorded for a specific version of a vocabulary.
Pre-conditions	Historical information has been imported from native format to the LexBIG repository.
Flow of Events	1) Actor invokes LexBIG API through client program specifying code system / version 2) System queries available history based on the given parameters
Post Conditions	Actor gets results (as specified by the LexBIG API) from the query.
Notes	

3.4.3.3 Determine State of Individual Concept

Use Case Id	**HST_UC_03**
Primary Actor	LexBIG User
Secondary Actors	Client Program
Brief Description	Queries current state of a given concept.
Pre-conditions	
Flow of Events	1) Actor invokes LexBIG API through client program specifying concept ID and property names that correspond to the desired status (e.g. concept status, isActive flag, etc) 2) System queries assigned properties for the concept
Post Conditions	Actor gets results (as specified by the LexBIG API) from the query.
Notes	

3.4.3.4 Retrieve Available Versions of a Code System

Use Case Id	**HST_UC_04**
Primary Actor	LexBIG User
Secondary Actors	Client Program
Brief Description	Actor invokes the LexBIG API to query versions of a vocabulary stored in the LexBIG repository.
Pre-conditions	
Flow of Events	1) Actor invokes LexBIG API through client program specifying code system name. 2) System queries accessible versions of the vocabulary.
Post Conditions	Actor gets results (as specified by the LexBIG API) from the query.
Notes	

3.5 Security, Integrity, and Access Constraints

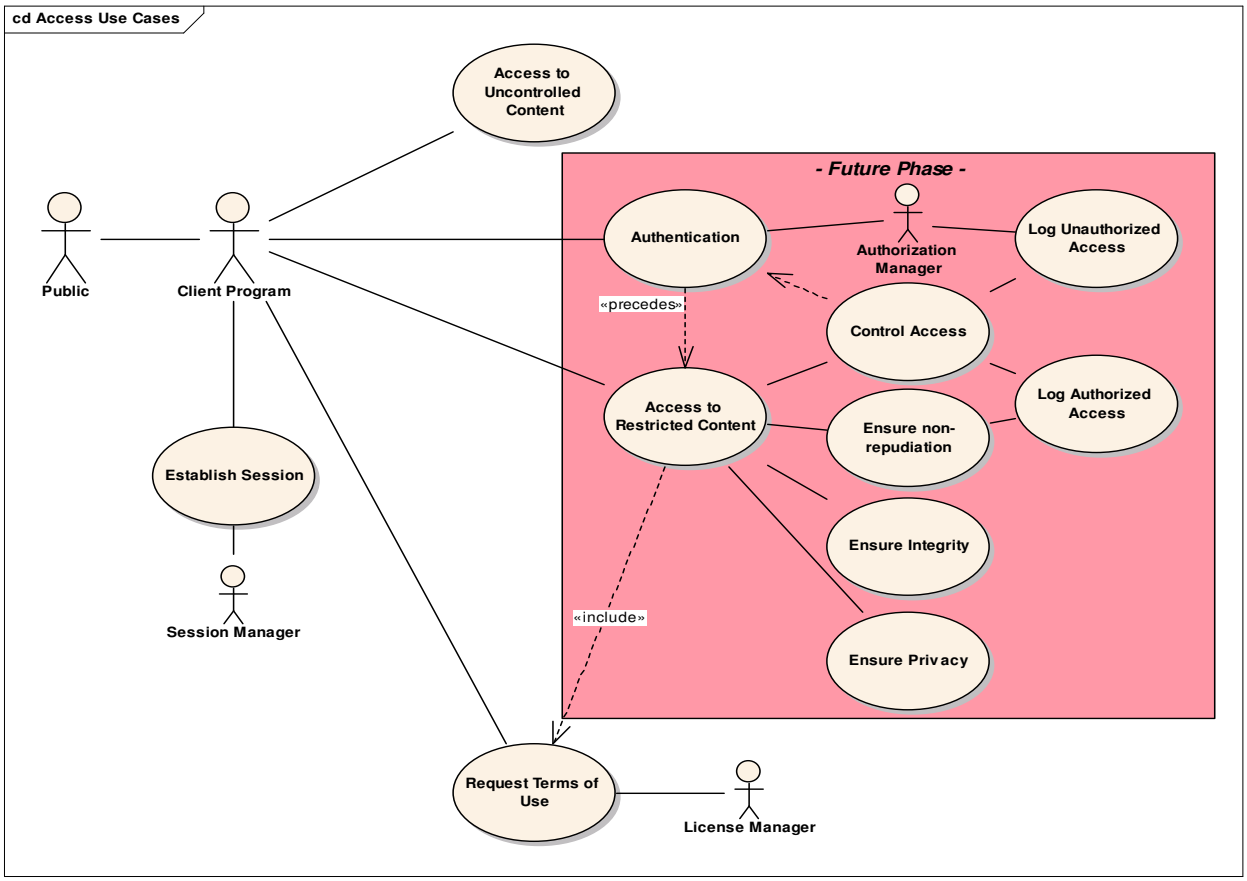
This section describes measures taken at runtime to ensure security and integrity for protected content. To support the goals of vocabulary federation the vocabulary server may need to protect some vocabulary content from unauthorized use. License and terms of usage are dependent on the vocabulary provider. Ultimately the license and terms of usage must be understood by the organization using the vocabulary. The vocabulary server will provide some mechanism for supporting security policy.

In consideration of baseline requirements, this document describes support of security policy in terms of active and passive models. In the passive model the server provides access to license terms as specified by the content provider. However, the server itself does not attempt to monitor or deny access based on this information. Instead, responsibility for appropriate access is to be delegated to the client application and end user. Client programs that directly interact with the user are expected to monitor and present license/copyright agreements when requests are made against protected content. In this situation, a burden is also placed on the user to properly interpret and honor the terms once disclosed.

In contrast, the active model implies a more traditional token-based (e.g. password) authentication model. If an unauthorized user ID or token is presented on a request, authentication fails and any attempt to access protected content is denied outright by the server.

Phase 1 deliverables for the LexBIG project will incorporate support only for the passive security model. However, active authentication is also considered in the use case diagram so that the current design does not preclude its introduction in the future. Additional use cases will be added in the final draft to further describe both models and anticipated support within the LexBIG runtime environment.

The notion of a bound session is potentially utilized in either scenario.



3.5.1 Request Terms of Use

Use Case ID	SEC_UC_01
Primary Actor	Public
Secondary Actors	Client Program, License Manager
Brief Description	System handles query of license terms in support of passive security model.
Pre-conditions	System configured in support of passive security model. Terms of use have been acquired from the content provider and registered for programmatic access.
Flow of Events	1) Actor initiates request to access a vocabulary with restricted license terms through a client program 2) Client program requests terms of use for the vocabulary from the system. 3) System consults license manager; retrieves and returns terms to client application. 4) Client application evaluates response. If content is restricted, a dialog is displayed to the user calling out the terms and requesting confirmation to continue. 5) User interprets license terms and confirms or cancels the request.
Post Conditions	If the action is confirmed, requested information will be returned to the user. Otherwise, the client program returns nothing and awaits the next request.
Notes	

3.6 Collaboration

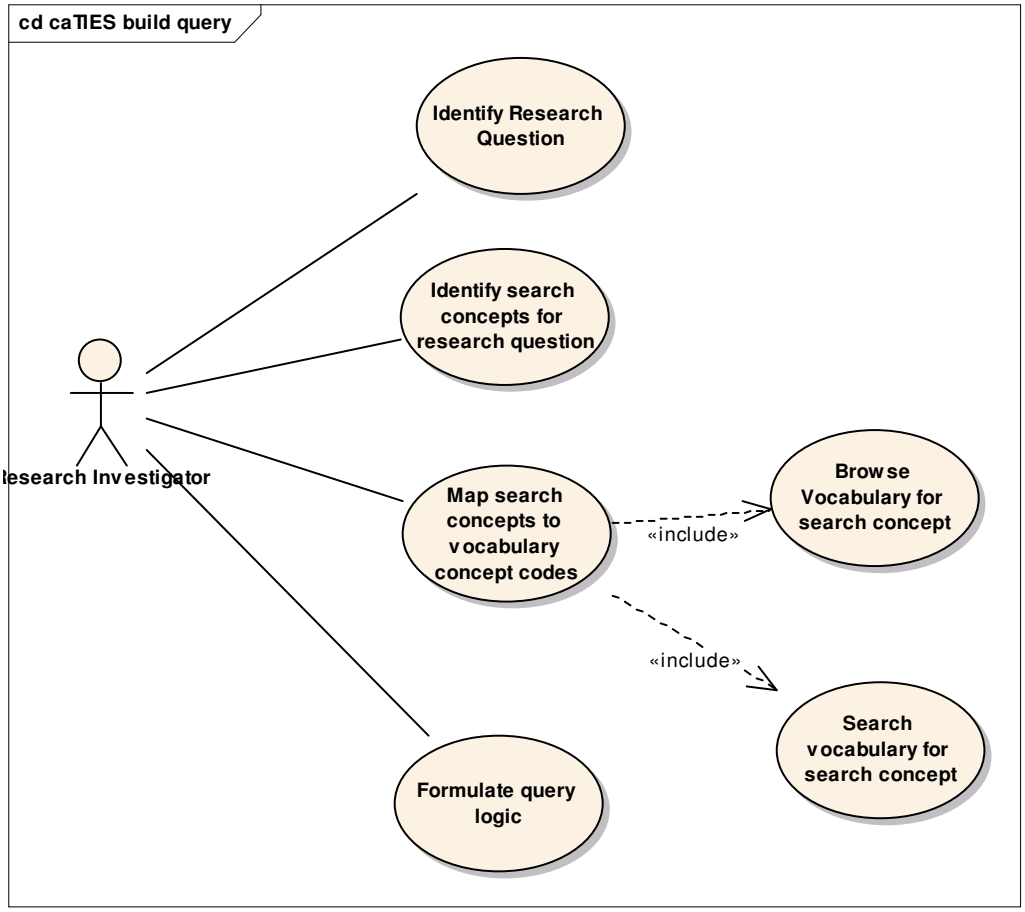
This section covers use cases derived based on input from key organizations or initiatives within the caBIG community working in association with the LexBIG project.

3.6.1 caTIES Build Query

A research investigator (user) is formulating search criteria to support a research question. The search criteria can represent a single search condition or a complex Boolean expression of unions, intersections, or exceptions. To build a query condition the caTIES application supports two modes of creation.

The user can browse the vocabulary to identify concepts of interest. Based on the type of vocabulary content explicit relationships may be navigable to aid the user in finding the query.

Alternately, the user can enter a search expression. This search expression can be checked for spelling and used to search the vocabulary to identify and appropriate matching concept. Searching the vocabulary based on the search expression will look for exact and lexically similar concepts based on the description of the concept. Once appropriate vocabulary concepts have been identified the user can assemble the concepts into a query expression using standard Boolean logic.



3.6.1.1 Identify Research Question

Use Case ID	caTIES_UC_01
Primary Actor	Research Investigator
Secondary Actors	NA
Brief Description	Actor formulates a research question based on research hypothesis and research dilemma
Pre-conditions	
Flow of Events	1) Identify research dilemma 2) Review, explore, and determine related or prior knowledge 3) Define the research hypothesis 4) Review, explore, and determine related or prior knowledge 5) Formulate a set of research questions 6) Formulate s set of retrieval parameters to fulfill research question
Post Conditions	A set of research questions and retrievals to fulfill the answer or partial answer to the research question.
Notes	

3.6.1.2 Identify search concepts for data retrieval

Use Case ID	caTIES_UC_02
Primary Actor	Research Investigator
Secondary Actors	NA
Brief Description	Actor determines the medical concepts required to fulfill the data retrieval request
Pre-conditions	Knowledge of vocabulary scheme used for source data
Flow of Events	1) Identify subject areas for research question 2) Identify search concepts in subject area 3) Determine the concept granularity requirements
Post Conditions	A list of search concepts
Notes	

3.6.1.3 Map search concepts to vocabulary concept codes

Use Case ID	caTIES_UC_03
Primary Actor	Research Investigator
Secondary Actors	NA
Brief Description	Actor determines the mapping of search concepts to appropriate vocabulary concepts or data elements
Pre-conditions	Knowledge of vocabulary concepts used for source data
Flow of Events	1) For each search concept identify vocabulary concept(s) 2) Identify concept specificity for search concept 3) Find vocabulary concept(s) representing search concept
Post Conditions	A list of vocabulary concepts
Notes	

3.6.1.4 Browse vocabulary for search concept

Use Case ID	caTIES_UC_04
Primary Actor	Research Investigator
Secondary Actors	NA
Brief Description	Actor browses vocabulary contents to find the concept(s) representing search concept.
Pre-conditions	Actor has some domain knowledge of subject matter of vocabulary
Flow of Events	1) Actor reviews the content of the vocabulary 2) Actor traverse navigitable relationships to aid in find vocabulary concept

	3) Actor selects vocabulary concept(s) that match search concept
Post Conditions	A list of vocabulary concepts
Notes	

3.6.1.5 Search vocabulary for search concept

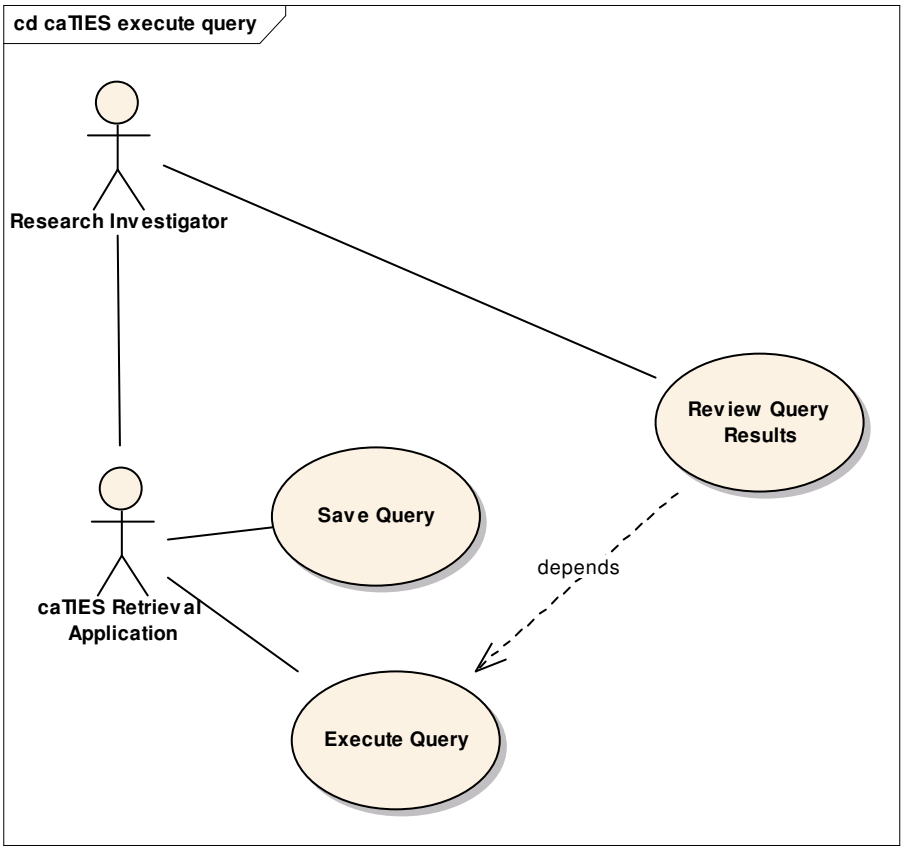
Use Case ID	**caTIES_UC_05**
Primary Actor	Research Investigator
Secondary Actors	NA
Brief Description	Actor applies logic operators to the search conditions represented by vocabulary concepts
Pre-conditions	Actor has some familiarity of Boolean logic
Flow of Events	1) Actor selects vocabulary concepts to be grouped for a Boolean function (and, or, not) 2) Actor repeats grouping and logic until search query matches retrieval question 3) Actor selects vocabulary concept(s) that match search concept
Post Conditions	A query
Notes	

3.6.1.6 Formulate query logic

Use Case ID	**caTIES_UC_06**
Primary Actor	Research Investigator
Secondary Actors	NA
Brief Description	Actor enters a search expression (word or phrase) represent the lexical form of the vocabulary concept description.
Pre-conditions	Actor has some domain knowledge of subject matter of vocabulary
Flow of Events	1) Actor enters search expression in a the form of words representing lexical form of concept description 2) Actor reviews vocabulary concepts the match the search expression 3) Actor selects vocabulary concept(s) that match search concept
Post Conditions	A list of vocabulary concepts
Notes	

3.6.2 caTIES Execute Query

This use case is initiated by the research investigator once a retrieval query has been created. The research investigator submits the query to the caTIES system. Once the query is completed, the investigator reviews the results (documents) for appropriate level of detail and correctness as required of the research question. If necessary, the investigator to modify or refine query to retrieve for applicable results. The investigator can save the query with search conditions for later reference and use.



3.6.2.1 Execute Query

Use Case ID	caTIES_UC_07
Primary Actor	Research Investigator
Secondary Actors	caTIES Retrieval Application
Brief Description	Actor used retrieval application to submit query to caTIES system.
Pre-conditions	Query has been defined.
Flow of Events	1) Submit query to caTIES system 2) Review acknowledgement from system
Post Conditions	Acknowledgement of query submission
Notes	

3.6.2.2 Save Query

Use Case ID	caTIES_UC_07
Primary Actor	Research Investigator
Secondary Actors	caTIES Retrieval Application
Brief Description	Actor saves a query to the caTIES retrieval environment for later reference, use, or refinement.
Pre-conditions	Query has been defined.

Flow of Events	1) Save query 2) Review save acknowledgement
Post Conditions	Saved query
Notes	

3.6.2.3 Review Query results

Use Case ID	**caTIES_UC_08**
Primary Actor	Research Investigator
Secondary Actors	caTIES Retrieval Application
Brief Description	Actor review the query results (documents) to determine appropriateness of query definition as relates to the research question.
Pre-conditions	Query has been defined and submitted to caTIES system
Flow of Events	1) Query results are displayed 2) Actor reviews results for relevance
Post Conditions	None
Notes	

4 Sign Off

4.1 Approval

Workspace General Contractor Rep	*Print Name*	*Date*
Architecture Workspace Rep	*Print Name*	*Date*
VCDE Workspace Rep	*Print Name*	*Date*
Workspace Working Group Rep	*Print Name*	*Date*
NCICB Rep	*Print Name*	*Date*